Session J: Finite Difference Solutions of Differential Equations, Part 3.

J.1 Introduction

Another technique of solving initial value problems involving differential equations, which is particularly powerful using digital computers, is the **Runge-Kutta** method. An initial value problem is one where you know the value and derivative of a function at a specific point. You can then advance the solution from the initial point $x = x_0$ to $x = x_0 + h$, for example, by applying Euler's approximation:

$$y_{n+1}(x_0 + h) \approx y_n(x_0) + hy'_n(x_0)$$

This amounts to considering the first two terms in the Taylor expansion of y about x_0. The obvious drawback with this technique is that it only uses information of the derivative at the initial point. Better accuracy can be obtained by keeping higher order terms (higher derivatives), but there is another, more elegant, solution. We can improve the accuracy of our solution by evaluating the gradient of the function at several points in between our initial and final values to obtain an *average* of the gradient across the interval. This is more elegant because we only need the first derivative of the function. The tricky part is deciding where to evaluate the derivative and what weight to give them. In this exercise, the aim is to introduce you to the basic principles of the Runge-Kutta method. The algebra gets a little hairy in places, but simplifies to a rather intuitive answer, so bear with it!

J.2 The Runge-Kutta Method

Assume that we have a differential equation, where the value of the function, y, is known at an initial point, x_0:

$$\frac{dy}{dx} = y'(x) = f(x, y) \quad \text{where} \quad y(x_0) = y_0 \tag{J.1}$$

In the following remember that $f(x, y)$ refers to the *derivative* and not the function itself. Now, suppose we consider the Taylor expansion of our *function* about a general point$x = x_n$:

$$y_{n+1} \approx y_n + hy'_n + \frac{h^2}{2!}y''_n + \frac{h^3}{3!}y'''_n \dots \tag{J.2}$$

where $y_n = y(x_n)$. As I mentioned in the introduction, the first two terms gives us Euler's approximation to the solution. In the Runge-Kutta method, better accuracy is achieved, not by keeping the higher order terms, but by replacing this series with the approximation:

$$y_{n+1} \approx y(x_n) + h \begin{bmatrix} \alpha_0 f(x_n, y_n) \\ +\alpha_1 f(x_n + \mu_1 h, y_n + \lambda_1 h) \\ +\alpha_2 f(x_n + \mu_2 h, y_n + \lambda_2 h) \\ + \dots \\ +\alpha_p f(x_n + \mu_p h, y_n + \lambda_p h) \end{bmatrix} \tag{J.3}$$

This looks terribly complicated, but all we are saying is that we will evaluate the derivative at several intermediate points within the step h. The coefficients μ_i and λ_i determine the fraction of the step we use, and the coefficients α_i determine the weight we give them. We must now find their values. Let's consider the simplest case where $p = 1$, which means we'll calculate the derivative at the initial point and one intermediate point. Since we now only have μ_1 and λ_1 to worry about, we shall drop the subscripts and just write μ and λ.

We can write equation J.3 in a slightly different form:

$$y_{n+1} \approx y_n + \alpha_0 k_0 + \alpha_1 k_1 \tag{J.4}$$

where k_0 and k_1 are given by:

$$\begin{gathered} k_0 = hf(x_n, y_n) \\ k_1 = h\left[f(x_n, y_n) + (\mu h f_x + \lambda k_0 f_y) + \tfrac{1}{2}(\mu^2 h^2 f_{xx} + 2\mu\lambda h k_0 f_{xy} + \lambda^2 k_0^2 f_{yy}) + O(h^3)\right] \end{gathered} \tag{J.5}$$

This is the slightly hairy bit that I promised you. In fact, k_1 is quite straightforward to obtain and is nothing more than a two-dimensional Taylor series expansion[1]. To simplify the notation a little, I've used:

$$\begin{gathered} f_x = \frac{\partial}{\partial x} f(x, y)\bigg|_{x_n, y_n} \qquad f_y = \frac{\partial}{\partial y} f(x, y)\bigg|_{x_n, y_n} \\ f_{xx} = \frac{\partial^2}{\partial x^2} f(x, y)\bigg|_{x_n, y_n} \qquad f_{xy} = f_{yx} \frac{\partial^2}{\partial x \partial y} f(x, y)\bigg|_{x_n, y_n} \qquad f_{yy} = \frac{\partial^2}{\partial y^2} f(x, y)\bigg|_{x_n, y_n} \end{gathered} \tag{J.6}$$

Using the definition of k_0 and writing $f(x, y)$ as f, we can simplify the expression for k_1 further:

$$k_1 = hf + h^2(\mu f_x + \lambda f f_y) + \frac{h^3}{2}(\mu^2 f_{xx} + 2\mu\lambda f f_{xy} + \lambda^2 f^2 f_{yy}) + O(h^4)$$

Substituting into equation J.4 we can write our solution as:

$$\begin{gathered} y_{n+1} = y_n + h(\alpha_0 + \alpha_1) f + h^2 \alpha_1 (\mu f_x + \lambda f f_y) + \\ \frac{h^3 \alpha_1}{2}(\mu^2 f_{xx} + 2\mu\lambda f f_{xy} + \lambda^2 f^2 f_{yy}) + O(h^4) \end{gathered} \tag{J.7}$$

Don't worry that I've now introduced second order derivatives. The reason why I have expanded the approximate solution (J.4), which only uses information about the gradient, is to compare what we get with the full Taylor expansion of the function given in J.2. Rewriting J.2 in the same notation, we find that:

$$\begin{gathered} y_{n+1} = y_n + hf + \frac{h^2}{2}(f_x + f f_y) \\ + \frac{h^3}{6}(f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2) + O(h^4) \end{gathered} \tag{J.8}$$

In order to show this, you need to be able to differentiate a function of two variables. This requires the application of the chain rule:

$$\begin{aligned} y'_n &= \frac{dy}{dx} = f(x, y) \equiv f \\ y''_n &= \frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x} \equiv f_x + f_y f \\ y'''_n &= \frac{d^2 f}{dx^2} = \dots \end{aligned}$$

[1] see, for example, http://mathworld.wolfram.com/TaylorSeries.html

By comparing equations J.7 and J.8 we can now see that the coefficients are not completely free parameters. Comparing terms up to h^2 you can see that they are constrained by the following relationships:

$$(\alpha_0 + \alpha_1) = 1, \quad \mu\alpha_1 = \frac{1}{2}, \quad \lambda\alpha_1 = \frac{1}{2}$$
$$\text{or} \tag{J.9}$$
$$\alpha_0 = 1 - \alpha_1, \quad \mu = \frac{1}{2\alpha_1}, \quad \lambda = \frac{1}{2\alpha_1}$$

What we are saying is that there is, in fact, only one *arbitrary* non-zero constant, namely α_1. Writing everything in terms of $\alpha = \alpha_1$ (since we only have one of them), equation J.7 becomes:

$$\begin{aligned} y_{n+1} &= y_n + hf + \frac{h^2}{2}(f_x + ff_y) \\ &+ \frac{h^3}{8\alpha}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + O(h^4) \end{aligned} \tag{J.10}$$

Note that the difference in the h^3 term simply indicates that the Runge-Kutta solution is not completely equivalent to the Taylor expansion at this order. However, the advantage is that it only uses information about the gradient of the function (J.3) and not higher orders. We only expanded $k_1 = f(x_n + \mu_1 h, y_n + \lambda_1 h)$ to show they were roughly equivalent and to constrain the seemingly large number of free parameters.

Although the choice of α is arbitrary, experience shows that $\alpha = 0.5$ is a good choice. Notice that this gives $\mu = \lambda = 1$, which means that we evaluate the derivative at the beginning and end of the interval (compare with equation J.3). (If we'd chosen $\alpha = 1$, then this would have corresponded to evaluating the derivative at the midpoint only). Because we evaluate the derivative at two points, this is known as a second order Runge-Kutta method. Our solution is therefore:

$$y_{n+1} = y_n + \frac{1}{2}(k_0 + k_1) \tag{J.11}$$

where

$$\begin{aligned} k_0 &= hf(x_n, y_n) \\ k_1 &= hf(x_n + h, y_n + k_0) \end{aligned} \tag{J.12}$$

or, if you prefer:

$$y_{n+1} = y_n + \frac{h}{2}f(x_n, y_n) + \frac{h}{2}f\left(x_n + h, y_n + hf(x_n, y_n)\right) \tag{J.13}$$

Close examination of J.12 shows that what we are doing is the following. We use the gradient at the initial point to calculate an approximate solution at the end point (Euler's method). We then calculate the gradient at this approximate solution and use both values in equation J.11. Hence, we are effectively using an approximation to the average gradient to obtain the solution.

Exercise J.1

1. Given $\dfrac{dy}{dx} = x^3 - y$ with $y(x_0) = y_0 = 1$ at $x_0 = 0$, find values of y_1 and y_2 where $h = 1/10$. By applying equation J.13 you should be able to show that:

$$y_1 = y_0 + \frac{1}{20}\left(x_0^3 - y_0\right) + \frac{1}{20}\left[\left(x_0 + \frac{1}{10}\right)^3 - \left\{y_0 + \frac{1}{10}\left(x_0^3 - y_0\right)\right\}\right]$$

and hence, that $y_1 = 0.90505$.

2. Now progress forward and find y_2 from y_1. You should find that $y_2 = 0.819515$.

   The error is of the order of of h^3, or 10^{-3} in this case, so your answers are accurate to about the third decimal place. Remember that this example has been done with $p = 1$. In general, calculating more intermediate values of the derivative will give more accurate results.

Now try the following example for yourself, again using the second order Runge-Kutta method given in equation J.13.

Exercise J.2

Given $\dfrac{dy}{dx} = e^{-x} - y$ with $y(x_0) = y_0 = 0$ at $x_0 = 0$, find values of y_1 and y_2 where $h = 1/10$.

you should find that $y_2 = 0.1633$.

J.3 The fourth-order Runge-Kutta method

Of course, we can obtain higher accuracy by going to higher order. In the Runge-Kutta method, this means evaluating the gradient at more intermediate points. However, most common applications of the Runge-Kutta method stop at fourth order (use three intermediate evaluations of the gradient). Fourth order Runge-Kutta methods are extremely powerful and have many applications. Below, I quote without proof a common expression of a fourth order Runge-Kutta method. It actually comes from Margenau and Murphy's book, *The Mathematics of Physics and Chemistry* (Van Nostrand 1964). An equivalent version can be found in *Numerical Recipes* by W. H. Press *et al.* In the following it has been slightly modified in nomenclature to correspond to the second order method derived above.

Given that we have a differential equation with a known initial value:

$$\frac{dy}{dx} = y'(x) = f(x, y) \quad \text{where} \quad y(x_0) = y_0 \tag{J.14}$$

then, to advance the solution from $x = x_0$ to $x = x_0 + h$, calculate:

$$y_1 = y_0 + \Delta y$$

where

$$\Delta y = \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3) \tag{J.15}$$

and

$$\begin{aligned} k_0 &= hf(x_0, y_0) \\ k_1 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_0}{2}\right) \\ k_2 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) \\ k_3 &= hf(x_0 + h, y_0 + k_2) \end{aligned}$$

Fourth order Runge-Kutta methods are accurate to order h^5. Interestingly, if $f(x, y)$ is independent of y, equation J.15 simply reduces to Simpson's Rule.

Exercise J.3

Repeat Exercise J.2 but now using equation J.15.

J.4 Systems of coupled differential equations

There are several instances where we need to solve several differential equations simultaneously. For example, the concentration of the neutron poison xenon in a reactor depends on the concentration of iodine - so their rates of change are linked together. Similarly, the point kinetics equations describe the rate of change of neutron density in terms of production by fission and delayed neutron precursor concentrations, which in turn depend on the neutron density...

The Runge-Kutta machinery described above can be extended in a straightforward way for these cases. For example, if we have two coupled first order differential equations with known initial values:

$$\begin{aligned}\frac{dy}{dx} &= y'(x) = f(x, y, z) \quad \text{where} \quad y(x_0) = y_0 \\ \frac{dz}{dx} &= z'(x) = g(x, y, z) \quad \text{where} \quad z(x_0) = z_0\end{aligned}$$

where changes in y influence z and vice versa. We can advance the solution of both equations from $x = x_0$ to $x = x_0 + h$ using:

$$y_1 = y_0 + \Delta y \qquad z_1 = z_0 + \Delta z$$

where

$$\begin{aligned}\Delta y &= \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3) \\ \Delta z &= \frac{1}{6}(q_0 + 2q_1 + 2q_2 + q_3) \qquad \text{(J.16)}\end{aligned}$$

and

$$\begin{aligned}k_0 &= hf(x_0, y_0, z_0) & q_0 &= hg(x_0, y_0, z_0) \\ k_1 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_0}{2}, z_0 + \frac{q_0}{2}\right) & q_1 &= hg\left(x_0 + \frac{h}{2}, y_0 + \frac{k_0}{2}, z_0 + \frac{q_0}{2}\right) \\ k_2 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}, z_0 + \frac{q_1}{2}\right) & q_2 &= hg\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}, z_0 + \frac{q_1}{2}\right) \\ k_3 &= hf(x_0 + h, y_0 + k_2, z_0 + q_2) & q_3 &= hg(x_0 + h, y_0 + k_2, z_0 + q_2)\end{aligned}$$

The extension to more equations in more variables should be obvious.

J.5 Second order equations

We have looked at advancing solutions of first order differential equations, but the techniques above can be used to find the solutions of second order differential equations as well. This can be achieved by noting that we can always split a second order differential equation into two first order differential equations. For example, in thermal hydraulics, we want to know the temperature distribution across a fuel rod. The radial temperature distribution is described by the following:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{dT}{dr}\right) = -\frac{u}{k}$$

where T is the temperature at radius r, u is the heat generated per unit volume and k is the coefficient of thermal conductivity.

This can be split into the two first order equations:

$$\begin{aligned}\frac{dT}{dr} &= z(r)\\ \frac{dz}{dr} &= -\frac{u}{k} - \frac{z}{r}\end{aligned}$$

which can be solved for T (and z) using the Runge-Kutta method in equations J.16.

Exercise J.4

Consider a fuel rod with cladding of thickness 1 mm and inner diameter 10 mm. In this case, there is no heat being generated in the cladding, so $u = 0$. If the temperature of the inner surface of the cladding is T_0 and the heat passing through unit length of the cladding is R, then the boundary conditions are:

$$T(r = r_0) = T_0, \qquad \frac{dT}{dr}(r = r_0) = z(r_0) = -\frac{R}{2\pi k r_0}$$

where r_0 is the radius of the inner surface of the cladding.

Using the system of equations given above, and values of $T_0 = 360°\mathrm{C}$, $k = 20\mathrm{Wm}^{-1}\mathrm{K}^{-1}$ and $R = 26.5\ \mathrm{kWm}^{-1}$, use the fourth order Runge-Kutta equations (J.16) to advance the solution from $r_0 = 5$ mm in steps of 0.1 mm to determine the temperature throughout the cladding to the outer surface.

Note: Think about the layout of your spreadsheet to minimize the amount of typing you have to do. Remember that selecting a cell or range of cells, and then dragging the selection will automatically update any references to other cells (unless absolute references like B3 are used). For example, suppose cell C3 contains '= A1*B3 − A3', selecting this cell and dragging down to row 10 would fill in cell C4 with '= A1*B4 − A4' and so on. You could then select all cells from C3 to C10, and dragging this selection to column D would result in D3 containing '= A1*C3 − B3', D4 with '= A1*C4 − B4' and so on.

This is the last Numerical Analysis session. A set of assessment exercises will be issued during the last week of term, to be completed a few weeks into the Spring Term (details to follow). The assessment will be based around several of the exercises that you should have already attempted, taken from several sessions. You are therefore encouraged to complete all the exercises on the various sheets that have been posted before.